Landscaping VR

An Honors Creative Project (HONR 499)

By

Joshua Schoen

Thesis Advisor

Dr. Paul Gestwicki

Ball State University
Muncie, Indiana

April 2018

Expected Date of Graduation

May 2018

Abstract

Virtual reality (VR) as a concept has been around for years, primarily through popular culture in science fiction novels and films such as *Snow Crash*, *The Matrix*, and even as early back as the 1930s with *The Man Who Awoke*. However, the technology has been too expensive for it to really become commonplace. Today we live in a society where we have developed technology enough to allow VR to be affordable and accessible enough to the public that applications for it are actually sought after. *Landscaping VR* is a dive into some of the possibilities afforded by the unique design space VR offers. There are two modes in the application. The editor mode, allows the user to create their own landscape design from a gods-eye view using the tools available. Once they are satisfied with their creation, they can shrink down into an exploration mode to see what their given landscape might look like from a human scale, then, conversely, grow back to the original size to alter their creation. It is a sandbox application in nature that takes advantage of motion control input and the unique perspective into virtual worlds.

Acknowledgments

I would like to thank Dr. Paul Gestwicki for not only advising me on this project, but also guiding me through my college career through the Computer Science department and the Honors College.

I would like to thank Dr. Scott Rice-Snow for the inspiration for this project.

I would like to thank the Digital Corps for helping guide my learning in different technologies and encouraging my pursuit into virtual and augmented reality.

I would like to thank my friends and family for helping and encouraging me to complete this last chapter in my college career.

Landscaping VR was a concept I had come up with during a course, *Inquiries in Earth Sciences*, held by Dr. Rice-Snow. The course delved into how different landscapes were formed and in what environments they might be found in order to more accurately portray the terrain for a scene in storytelling, game design, etc. It was intriguing. I have always been a gamer at heart and have taken up the mantle of gamemaster in many tabletop games before, but I had never delved so deeply into the scenery of where my players had been. I suddenly found that my games were more alive, and the players were taking advantage of the new details. Every cave was possible treasure, and every cliffside could offer more lands to explore. This began my interest into landscaping.

We were encouraged throughout the course to take the different land features and find a way to incorporate it into some form of creative outlet, particularly, though not limited to, something within our field of expertise. In one instance, we were looking at a massive area that had been carved out by a glacier. It was full of nooks and crannies, and I wondered what if one were able to get a digital copy of the area and then be able to shrink down to explore the crevices at a more finite detail than from our current perspective. This idea later evolved into a more interactive experience as I decided to create an application that could perhaps be utilized by landscape architects to build a scene of their design and then shrink down to view what their scene would look like in practicality.

Thus was the first stage of *Landscaping VR*. From a base layout, the user could sculpt the land into hills and valleys, add a variety of foliage (trees, bushes, etc) and

aquatic elements (rivers, lakes, etc), and paint different textures to resemble different terrains (rocky, grassy, etc). They could then shrink to view their creation, and grow again to alter it. Afterwards the user would be able to save their creation to look over or rework at a later time. I decided to use Unreal Engine 4 (UE4) to create the application and use the Oculus Rift to test it. I was just beginning to learn UE4 at the time and wanted to continue my exploration into its possibilities by working with its VR toolkit. The Oculus Rift is a set of VR hardware that I own, and thus would be easy to access for testing. These were the initial parameters I set for the project.

The first hurdle I had to overcome was the software I was trying to use. UE4 is a powerful game engine that can do a lot in the right hands, and I've worked with it before to create other games under the tutelage of Dr. Gestwicki. However, the VR toolkit was proving to be more than a handful. The player height and positioning wasn't being calculated correctly when forum posts and stackexchange sites claimed it should be done automatically by UE4. Furthermore, a couple of the features I was trying to work on were nigh impossible with the time and resources I had. Though because of my lack of experience with UE4, it took me far longer to realize that I couldn't do these features. I kept working through thinking the problem was my inexperience with UE4 rather than the feature itself. I ended up switching from UE4 to Unity. Unity is another game engine, powerful in its own right, but its VR/AR capabilities are very new and primarily require outside libraries. However, I've had much more experience working with Unity than UE4 and have even used Unity to work on VR projects before.

At this point, I realized that attempting to learn new software while also diving into new techniques to approach VR (the primary purpose of the project) was my main setback, so I wanted to approach the project from as familiar a point as I could. In my past experience with VR in Unity I used an OpenVR library known as SteamVR. OpenVR denotes that the library controls input and output for most VR hardware, so in theory it should work fine for the Rift which I was using. Unfortunately, this was not quite the case. Oculus works slightly differently from other VR systems. It has its own independent runtime events that, for full functionality, must be called separately from general OpenVR runtime events. Oculus does support Unity, but it would require downloading a new library which ran me into my initial problem of trying to learn on the fly. This I remedied by switching the system I tested on. I had access to an HTC Vive (another popular VR system) through the Digital Corps on Ball State University campus. The Vive is fully suited for OpenVR libraries, and thus the rest of the project could commence.

It was at this point, when I had eliminated as many hindering variables as I could, that I was able to look into the feasibility of my desired functionality. To mold the landscapes into the hills and valleys, as I had initially intended, required the ability to manipulate the polygons of the meshes I was using for my base layouts. A mesh in computer modeling is a series of polygons (usually triangles or squares) that form the shape of whatever object you are creating. However, these meshes are pretty hard set in their base coding. You can grow, shrink, move, and angle the object in any way you like, as long as the object retains its base figure. My initial concept for the application

would require breaking that figure, and that wasn't something I was able to feasibly do with my experience within the deadline of the project. Furthermore, I wasn't quite able to make out how to "paint" over the assets. I could change entire textures of assets, but altering a part of a texture was another piece of functionality that was just out of my expertise.

With both of these issues popping up as major roadblocks, and the deadline rapidly approaching, I opted for a slight redesign of the project. Instead of sculpting and painting a landscape, I decided to provide a set of landmass assets that the user could pick and choose from to set and place on a blank canvas. The process was far more accomplishable and worked out stunningly. It did however require more time to model actual assets to use for the project. I decided to relearn Blender, a free modeling application I've used lightly before. The decision to make my own assets was one I had made at the beginning of the project when the created assets were merely going to be some foliage and other decorative elements. I wanted assets that were part of a set, so they would flow well together. Any sets of assets online would have been far more expensive than I was willing to pay. Additionally, I preferred a more polygonal, cartoonish art style, so the creation of the assets was not terribly difficult.

I was going to limit the modeling to just a few hills of varying heights and shades, but as I worked on the modeling, I started envisioning mountain ranges, rolling hills, and grassy plains. I was already in the midst of a design shift, so I rolled with it. I created some larger, bolder landmasses and played around to see what it looked like. In the end, I liked what came of it. The application became a lot more large-scale. Instead of

designing parks and pathways, it models terrain and whole scenes. With this final transition, I had settled on a design for the application.

The final point of interest was the second half of the application's functionality: the shrinking. This ran into some interesting technical errors. The actual shrinking of the player was fairly simple enough, but the assets I used to indicate teleportation (my chosen mode of transportation while shrunken) remained at a normal size causing the teleportation indicator to look enormous while trying to move around. Furthermore, the render distance became an issue. Render distance determines how far and, as what became the issue, how close an object can be to the game camera before it stops rendering in the view. When on a normal scale, the usual issue to worry about is how far the render distance reaches. However when in a shrunken state, objects are now calculated far closer to the camera than before. Therefore, what ends up happening is that the ground beneath the player tends to vanish before their eyes. This particularly becomes an issue in VR, as it causes a serious amount of vertigo. In most games, disappearing objects simply looks sloppy, but in VR, your senses are more focused in on the virtual world surrounding you. When your eyes see that nothing is beneath where your feet are, it expects you to fall while you're still standing strong on the floor in the real world.

Obviously, this was quite an issue, as the shrinking was half of the goal of the project. The solution was actually far more elegant than trying to break the game engine and reduce the render minimum distance. If I can't shrink the player, then I just have to grow the terrain. The minimum render distance doesn't affect the player, because the

camera is at a normal range, and it appears as though the player has still shrunken into the landscape itself.

In the end, *Landscaping VR* became a sandbox application in which, from a base canvas, the user could place various landmass objects, including mountains, hills, and aquatic elements, to form various terrains for grassy, rocky, sandy, and snowy terrains. They could then shrink to view the creation and grow again to continue working on it. Unfortunately, due to time constraints, the ability to save the creation was removed from the scope.

Nonetheless, I would call the project an overall success. The two main components of the application were the creation of landscapes and the ability to shrink and view them from a "human" scale. This was to evoke imagination and a greater sense of perspective in the user. They're given a blank canvas and three dimensions to work on a landscape of their own design, whether they want to focus on a large mountain range, or perhaps a low and long river valley. Furthermore, they'll get the opportunity to view their brain-child from a whole new angle as they shrink down and see the mountains they placed rise high above them or the river flow low beneath them. They can wander the sandy hills to discover a lush oasis, or traverse the icy plains toward a lone mountain on the horizon. Whatever it is they wish to create, *Landscaping VR* can help manifest it into a whole new virtual reality.